

# Supplementary Material for: Designing microarray phantoms for hyperspectral imaging validation

Matthew L. Clarke,<sup>1,2</sup> Ji Youn Lee,<sup>1</sup> Daniel V. Samarov,<sup>3</sup> David W. Allen,<sup>4</sup>  
Maritoni Litorja,<sup>4</sup> Ralph Nossal,<sup>5</sup> and JeeSeong Hwang<sup>1,\*</sup>

<sup>1</sup>Radiation and Biomolecular Physics Division, Physical Measurement Laboratory, National Institute of Standards and Technology,  
Gaithersburg, MD 20899, USA

<sup>2</sup>Currently with the National Gallery of Art, Washington, DC, 20565, USA

<sup>3</sup>Statistical Engineering Division, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD  
20899, USA

<sup>4</sup>Sensor Science Division, Physical Measurement Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899, USA

<sup>5</sup>Eunice Kennedy Shriver National Institute of Child Health and Human Development, National Institutes of Health, Bethesda, MD USA 20892  
USA

\*[jeeseong.hwang@nist.gov](mailto:jeeseong.hwang@nist.gov)

## Example MATLAB code

```
function Spocle_Example_3DyeMix_Print_1slide
%collect all variables
title='3DyeMix_Example_Output'; %This is the output filename;
ColumnOffset=0; %use 0 here
RowOffset=0; %use 0 here
ColumnSpacing=0.00025; %spacing between spots in meters
RowSpacing=0.00025; %spacing between spots in meters
StartColumn=15; %This is the offset from the corner of the slide
StartRow=60; %This is the offset from the corner of the slide
PinCol=1; %A single pin should be placed in position 1, 1
PinRow=1; %A single pin should be placed in position 1, 1
NSlides=1; %# of slides, printing will start on slide 1 and print...
%through this #

NArrays=1; %#of same array shapes using SAME wells
ArrayOffset=60; %#row spaces between array starting positions...
%(should include array size)
WashSpots=150; %wash the pin after this # spots (excludes preprint)
Preprint.Spots=25; %use 0 if no preprinting desired
Preprint.Slide=1; %can preprint on same slide as sample
Preprint.StartColumn=5; %offset for preprint dots...
%(make sure does not overlap array)
Preprint.StartRow=7; %offset for preprint dots
Preprint.CurrentColumn=Preprint.StartColumn; %Sets preprint start
Preprint.CurrentRow=Preprint.StartRow; %Sets preprint start
Preprint.Limit=round(0.0254/ColumnSpacing)-10; %Sets end of preprint

%generate strings from variables
strColumnOffset=num2str(ColumnOffset);
strRowOffset=num2str(RowOffset);
strColumnSpacing=num2str(ColumnSpacing);
strRowSpacing=num2str(RowSpacing);

%begin creating output file
fid = fopen([title, '.spo'], 'wt'); %open new text file,...
%overwriting an existing one if needed
```

```

%create header
header=cell(5,1);
header{1}=['<Procedure
xmlns="http://www.korteks.com/spotbot/"><Version>1.1</Version><Title>',...
    title, '</Title>'];
header{2}=['<SlideOriginOffset><ColumnOffset>',strColumnOffset,...
    '</ColumnOffset><RowOffset>',strRowOffset,...
    '</RowOffset><SlideOriginOffset>'];
header{3}=['<SpotSpacing><ColumnSpacing>',strColumnSpacing,...
    '</ColumnSpacing><RowSpacing>',strRowSpacing,...
    '</RowSpacing><SpotSpacing>'];
header{4}='<ZAxisInhibit>False</ZAxisInhibit>';
header{5}=['<Lights><Power>True</Power><Blink>No</Blink></Lights><PinLoad><Type>SMP3</Type><Column>',...
    num2str(PinCol), '</Column><Row>', num2str(PinRow), ...

    '</Row></PinLoad><Lights><Power>False</Power><Blink>No</Blink></Lights>'];
for i=1:5
    fprintf(fid, '%-1s \n', header{i}); %cell is string
end

%load image data to convert to coordinates
I=imread('3DyeMixes-6pattern.tif'); %this is the image file with the array
wellcodes=dlmread('SpotBot 2010_09_3DyeMix well codes.txt','\t'); %This is
    %...a text file that links colors with well locations
keymap=wellcodes(:,3:5)/255; %key for colors, used to display image
I2=rgb2ind(I,keymap);
imshow(I2,keymap, 'InitialMagnification','fit'); %show keyed image
keyrows=wellcodes(:,1); %get rows
keycols=wellcodes(:,2); %get cols

%Now the program will look through the image, color by color, and determine
%the printing locations
for CurrSlide=1:NSlides
    %get spot coordinates for each "color"
    SpotCoor=cell(size(keymap,1),2);
    for i=1:size(keymap,1)
        if(keycols(i)==0)
            SpotCoor{i,1}=[]; %a well position of 0 indicates no...
                %printing of that color
            SpotCoor{i,2}=[];
        else
            [SpotCoor{i,1} SpotCoor{i,2}]=find(I2==i-1); %[row col]
        end
    end
end
for CurrArray=1:NArrays
    %create printing list
    if CurrArray<=4;
        RowAdjust=ArrayOffset*(CurrArray-1);
        ColAdjust=0;
    else
        RowAdjust=ArrayOffset*(CurrArray-5);
        ColAdjust=ArrayOffset;
    end
    for i=1:length(keycols)
        if(keycols(i)==0) %do not print

```

```

else %print
    Nspots=length(SpotCoor{i,1});
    WellRow=keyrows(i);
    WellCol=keycols(i);
    %wash pin
    wash_cycle(fid)
    wash_cycle(fid)
    %get the sample
    fprintf(fid, '%-1s \n',
['<GetSample><Number>1</Number><Column>',...
    num2str(WellCol), '</Column><Row>', num2str(WellRow), ...
    '</Row><Time>3</Time></GetSample>']);
    Preprint=preprint_spots(fid,Preprint);
    for j=1:Nspots
        fprintf(fid, '%-1s \n', ['<SpotSlide><Number>',...
            num2str(CurrSlide), '</Number><Column>',...
            num2str(SpotCoor{i,2}(j)+StartColumn+ColAdjust),...
            '</Column><Row>',...
            num2str(SpotCoor{i,1}(j)+StartRow+RowAdjust),...
            '</Row><Time>0</Time></SpotSlide>']);
        if(rem(j,WashSpots)==0 && j~=Nspots)
            %wash and get more sample
            wash_cycle(fid)
            fprintf(fid, '%-1s \n',
['<GetSample><Number>1</Number><Column>',...
            num2str(WellCol), '</Column><Row>',...
            num2str(WellRow), '</Row><Time>3</Time></GetSample>']);
            Preprint=preprint_spots(fid,Preprint);
        end
    end
end
end
end
end
end_wash(fid)

%end procedure
fprintf(fid, '%-1s
\n', '<Lights><Power>False</Power><Blink>No</Blink></Lights></Procedure>');
fclose(fid);

```

```

%This function is called above for preprinting
function Preprint=preprint_spots(fid,Preprint)
for i=1:Preprint.Spots
    if Preprint.CurrentColumn>=Preprint.Limit %column is full
        Preprint.CurrentColumn=5; %move to beginning of next row
        Preprint.CurrentRow=Preprint.CurrentRow+2;
    end
    fprintf(fid, '%-1s \n', ['<SpotSlide><Number>',...
        num2str(Preprint.Slide), '</Number><Column>',...
        num2str(Preprint.CurrentColumn), '</Column><Row>',...
        num2str(Preprint.CurrentRow), '</Row><Time>0</Time></SpotSlide>']);
    Preprint.CurrentColumn=Preprint.CurrentColumn+1;
end

```

```

%This function is called above to code the pin washing cycle
function wash_cycle(fid)
for i=1:3
    if(i<3)
        fprintf(fid, '%-1s \n', '<Wash><Time>0.5</Time></Wash>');
        fprintf(fid, '%-1s \n', '<Dry><Time>0.5</Time></Dry>');
    else
        fprintf(fid, '%-1s \n', '<Wash><Time>2</Time></Wash>');
        fprintf(fid, '%-1s \n', '<Dry><Time>10</Time></Dry>');
    end
end
end

```

```

%This function is called above to code the pin end washing cycle
function end_wash(fid)
for i=1:9
    if(i<9)
        fprintf(fid, '%-1s \n', '<Wash><Time>0.5</Time></Wash>');
        fprintf(fid, '%-1s \n', '<Dry><Time>0.5</Time></Dry>');
    else
        fprintf(fid, '%-1s \n', '<Wash><Time>2</Time></Wash>');
        fprintf(fid, '%-1s \n', '<Dry><Time>10</Time></Dry>');
    end
end
end

```